



LAWRENCE
LIVERMORE
NATIONAL
LABORATORY

UCRL-TR-237220

Equation of State Scalings in Kull

M. S. Ulitsky, G. Zimmerman, P. Renard, N. A. Tang

December 11, 2007

Disclaimer

This document was prepared as an account of work sponsored by an agency of the United States government. Neither the United States government nor Lawrence Livermore National Security, LLC, nor any of their employees makes any warranty, expressed or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States government or Lawrence Livermore National Security, LLC. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States government or Lawrence Livermore National Security, LLC, and shall not be used for advertising or product endorsement purposes.

This work performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344.

UCRL-TR-????

Equation of State Scalings in Kull

Mark Ulitsky, George Zimmerman, Paul Renard, and Nu Ai Tang
Lawrence Livermore National Laboratory

Contents

1	EOS Scalings	3
2	Creating an EOS Object with EOS Scalings	5

1 EOS Scalings

In this short report, we consider two types of equation of state (EOS) scalings, which will be referred to as type I (or density scaling) and type II (or Thomas-Fermi scaling). EOS scalings are used to correct for the fact that EOS tables are constructed with a fixed isotopic composition. Even if the isotopics in a computational zone are evolving through chemical or nuclear reactions, the EOS lookups for pressure and energy (and their derivatives with respect to density and temperature) aren't normally cognizant of this fact. The EOS scalings are an attempt to fix this shortcoming. They typically modify the incoming density and/or temperature based on ratios of isotopic quantities (like $\langle A \rangle_{table} / \langle A \rangle_{zone}$, $\langle Z \rangle_{table} / \langle Z \rangle_{zone}$), and then modify the table lookup values. In this way, the EOS can dynamically respond to the changing isotopics.

Type I is the simpler scaling and has the following form for the independent variables:

$$\rho' = \alpha \rho \quad (1)$$

$$T' = T, \quad (2)$$

where ρ' is the modified density used for the table lookup and α is given by

$$\alpha \equiv \frac{\langle A \rangle_{table}}{\langle A \rangle_{zone}} \frac{(\langle Z \rangle_{zone} + 1)}{(\langle Z \rangle_{table} + 1)}. \quad (3)$$

The table lookups then give modified values of pressure and specific energy (denoted by P' and E' respectively) which are related to the desired values of P and E by

$$P = P' \quad (4)$$

$$E = \alpha E'. \quad (5)$$

A slightly more involved scaling, referred to as type II, can be used when running the code in 3T mode. The scalings for density and temperature are now given by

$$\rho' = a \rho \quad (6)$$

$$T'_{ele} = b T_{ele} \quad (7)$$

$$T'_{ion} = b T_{ion}, \quad (8)$$

where a and b are given by

$$a \equiv \frac{\langle A \rangle_{table}}{\langle A \rangle_{zone}} \frac{\langle Z \rangle_{zone}}{\langle Z \rangle_{table}} \frac{\lambda_{table}^3}{\lambda_{zone}^3} \quad (9)$$

$$b \equiv \frac{\lambda_{table}^2}{\lambda_{zone}^2}, \quad (10)$$

and λ takes the form of

$$\lambda \equiv e^{\left(\frac{2}{3} \frac{\langle Z \ln Z \rangle}{\langle Z \rangle}\right)}. \quad (11)$$

Note that if the isotope list for a given material only involved a single value for Z , then λ would reduce to

$$\lambda = Z^{2/3}. \quad (12)$$

The scalings for the dependent variables are

$$P_{ele} = c_{ele} P'_{ele} \quad (13)$$

$$P_{ion} = c_{ion} P'_{ion} \quad (14)$$

$$E_{ele} = d_{ele} E'_{ele} \quad (15)$$

$$E_{ion} = d_{ion} E'_{ion}, \quad (16)$$

where the c and d coefficients are given by

$$c_{ele} \equiv \frac{\lambda_{zone}^5}{\lambda_{table}^5} \quad (17)$$

$$c_{ion} \equiv \frac{\lambda_{zone}^5 \langle Z \rangle_{table}}{\lambda_{table}^5 \langle Z \rangle_{zone}} \quad (18)$$

$$d_{ele} \equiv a c_{ele} \quad (19)$$

$$d_{ion} \equiv a c_{ion} . \quad (20)$$

Often we are interested in derivatives of pressure or specific energy with respect to density or temperature. A simple application of the chain rule from calculus provides the necessary formulas:

$$\frac{\partial P_{ele}}{\partial \rho} = \frac{\partial P_{ele}}{\partial \rho'} \frac{\partial \rho'}{\partial \rho} = a c_{ele} \frac{\partial P'_{ele}}{\partial \rho'} \quad (21)$$

$$\frac{\partial P_{ion}}{\partial \rho} = \frac{\partial P_{ion}}{\partial \rho'} \frac{\partial \rho'}{\partial \rho} = a c_{ion} \frac{\partial P'_{ion}}{\partial \rho'} \quad (22)$$

$$\frac{\partial E_{ele}}{\partial \rho} = \frac{\partial E_{ele}}{\partial \rho'} \frac{\partial \rho'}{\partial \rho} = a d_{ele} \frac{\partial E'_{ele}}{\partial \rho'} \quad (23)$$

$$\frac{\partial E_{ion}}{\partial \rho} = \frac{\partial E_{ion}}{\partial \rho'} \frac{\partial \rho'}{\partial \rho} = a d_{ion} \frac{\partial E'_{ion}}{\partial \rho'} \quad (24)$$

$$\frac{\partial P_{ele}}{\partial T} = \frac{\partial P_{ele}}{\partial T'} \frac{\partial T'}{\partial T} = b c_{ele} \frac{\partial P'_{ele}}{\partial T'} \quad (25)$$

$$\frac{\partial P_{ion}}{\partial T} = \frac{\partial P_{ion}}{\partial T'} \frac{\partial T'}{\partial T} = b c_{ion} \frac{\partial P'_{ion}}{\partial T'} \quad (26)$$

$$\frac{\partial E_{ele}}{\partial T} = \frac{\partial E_{ele}}{\partial T'} \frac{\partial T'}{\partial T} = b d_{ele} \frac{\partial E'_{ele}}{\partial T'} \quad (27)$$

$$\frac{\partial E_{ion}}{\partial T} = \frac{\partial E_{ion}}{\partial T'} \frac{\partial T'}{\partial T} = b d_{ion} \frac{\partial E'_{ion}}{\partial T'} . \quad (28)$$

2 Creating an EOS Object with EOS Scalings

To create an EOS object in a kull python script that makes use of EOS scalings is very straightforward. Basically, we need to add the keyword argument *isotopic scaling* to the EOS constructor, along with the value for this argument (choices are “density scaling” or “Thomas Fermi” or “None”). For example, if we want an EOS for SiO_2 (which is given by LEOS table 2210) to use the type I scaling, the correct syntax is

```
# Make a LEOS item with density-type EOS scaling
leosFactory = eos.LEOS(2210,
                       isotopicScaling="density scaling")
```

If we want the type II scaling for a 3T application, use

```
# Make a LEOS item with Thomas-Fermi EOS scaling
leosFactory = eos.LEOS(2210,
                       isotopicScaling="Thomas Fermi")
```

For some ICF applications, we may want to generate an atomic EOS that uses an EOS scaling. If we want to create an EOS for an atomic mixture of boron (LEOS number 50) and stainless steel (LEOS number 3010), we should do

```
leosFactoryB = eos.LEOS(50,
                       isotopicScaling="Thomas Fermi")

leosFactorySST = eos.LEOS(3010,
                       isotopicScaling="Thomas Fermi")

mixEOS = eos.MixEos([leosFactoryB, leosFactorySST],
                   [(0.0, 10.5), (10.5, 1000.0)],
                   equilibrateOption="TotalPressure")
```

Note that with an atomic or mix EOS, the tables should be in increasing “Z-order”. That is, the $\langle Z \rangle$ for boron is smaller than the $\langle Z \rangle$ for steel. Also, the second line of the mixEOS constructor indicates the “Z-ranges” that will be used to associate isotopes with each table. That is, isotopes with a Z between 0 and 10 will be associated with the boron table, while isotopes with higher Z-values will be associated with the steel table. The third line specifies the property that will be brought into equilibrium during the Newton-Raphson iteration in order to compute the sub-densities that are used for the individual table lookups. The possible choices for the equilibrateOption keyword are “TotalPressure”, “AnalyticElectronDensity”, “TabularElectronDensity”, and “ChemicalPotential”. Refer to UCRL-TR-224230, “Combining Equations of State in Kull” for more information on the atomic equation of state.